

**IN THE UNITED STATES DISTRICT COURT  
FOR THE DISTRICT OF DELAWARE**

CYBERFONE SYSTEMS, LLC,  Plaintiff,  v.  CELLCO PARTNERSHIP, ET AL.,  Defendants.	C.A. No. 1:11-cv-827-SLR
CYBERFONE SYSTEMS, LLC,  Plaintiff,  v.  AVAYA, INC., et al.,  Defendants.	C.A. No. 1:11-cv-830-SLR
CYBERFONE SYSTEMS, LLC,  Plaintiff,  v.  SONY ELECTRONICS, INC., et al.,  Defendants.	C.A. No. 1:11-cv-833-SLR
CYBERFONE SYSTEMS, LLC,  Plaintiff,  v.  FEDERAL EXPRESS CORP., et al.,  Defendants.	C.A. No. 1:11-cv-834-SLR

**PLAINTIFF CYBERFONE SYSTEMS, LLC'S SURREPLY BRIEF  
IN SUPPORT OF ITS EARLY CLAIM CONSTRUCTIONS**

**TABLE OF CONTENTS**

	<b>PAGE</b>
I. INTRODUCTION.....	1
II. ARGUMENT.....	2
A. Interaction Between “Form Driven Operating System” And “Transaction Assembly Server (TAS).” .....	2
B. “Form Driven Operating System.” .....	3
1. The Intrinsic Evidence Compels Cyberfone’s Proposed Construction.....	3
2. Defendants’ Proposed Construction Is Incomplete, Ignores The Specification, And Misconstrues The Intrinsic Record.....	6
C. “Client Module” Of ‘024 Patent. ....	9
D. “Transaction Assembly Server (TAS).” .....	11
1. The Intrinsic Evidence Compels Cyberfone’s Proposed Construction.....	12
2. Defendants’ Proposed Construction Does Nothing To Define TAS, Ignores The Specification And Is Inconsistent With Defendants’ Prior Position On The ‘024 Patent. ....	13
V. CONCLUSION.....	14

**TABLE OF AUTHORITIES**

**PAGE**

**Cases**

<i>Bicon Inc. v. Straumann Co.</i> 441 F.3d 945 (Fed. Cir. 2006) .....	13
<i>Johnson Worldwide v. Zebco</i> 175 F.3d 985 (Fed. Cir. 1999) .....	10
<i>Novartis Pharma. Corp. v. Abbott Labs.</i> 375 F.3d 1328 (Fed. Cir. 2004) .....	10
<i>Phillips v. AWH Corp.</i> 415 F.3d 1303 (Fed. Cir. 2005) .....	10
<i>Unique Concepts, Inc. v. Brown</i> 939 F.2d 1558 (Fed. Cir. 1991) .....	13

**Other Authority**

Patent Case Management Judicial Guide, Second Edition (2012) .....	10
--	----

## I. INTRODUCTION

In their attempt to seek constructions that will “necessarily lead to summary judgment of noninfringement” (Def. Op. Br. at 1), Defendants continue to ignore much of the intrinsic record, focus on an alleged *characteristic* of a form driven operating system (to the exclusion of addressing the actual functionality of a form driven operating system), and make no attempt at all to describe or define a transaction assembly server (or TAS).<sup>1</sup> In other words, Defendants reject claim construction as a process to give definition and meaning to the disputed terms, and simply offer positions with negative limitations to provide avenues from which they wish to argue non-infringement. Perhaps the clearest (and newest) example of this is Defendants’ rejection of their own position five weeks ago that “computer code for generating a data transaction” is equivalent to “transaction assembly server” (or TAS). Although this admission reinforces the conclusion that Cyberfone’s construction of TAS is correct, the remaining Defendants do not have products accused of infringing a claim that includes the “computer code for generating a data transaction” term of the ‘024 patent, and therefore have disavowed the statements they collectively made but five weeks ago. Defendants’ constructions, including their almost circular definition of TAS in conjunction with form driven operating system, are products of litigation and do not clarify the meaning of the claim terms for the jury or respect the inventions reflected therein.

Cyberfone’s constructions, on the other hand, address the actual inventions, the full specifications of the patents, and the prosecution histories as a whole. Cyberfone’s constructions, unlike Defendants’, adhere to the fundamental rules of claim construction, while giving substance to the claim terms being construed in this early proceeding. Cyberfone

---

<sup>1</sup> Cyberfone mistakenly referred to the TAS as a “terminal assembly server” in several places in its opening brief. This was obviously a typographical error, as TAS refers to a “transaction assembly server.” *See, e.g.*, Cyberfone’s Op. Br. at IV.D (entitled “Transaction Assembly Server (TAS).”). Cyberfone was not “gloss[ing] over the actual claimed term in dispute . . . .” (Def. Reply Br. at 12.)

respectfully requests that the Court adopt its proposed constructions of form driven operating system and transaction assembly server.

## II. ARGUMENT

### A. Interaction Between “Form Driven Operating System” And “Transaction Assembly Server (TAS).”

Although Defendants posit a tortured misreading of Cyberfone’s proposed construction of form driven operating system,<sup>2</sup> Cyberfone and Defendants **agree** that a form driven operating system consists of a transaction assembly server (TAS) working in connection with forms:

“As noted above, the **transaction assembly (application) server (TAS)** is a data stream stored in TAS PROM 95 which **together with the forms** from form/menu memory 96 **create a simple form driven operating system** which provides the necessary control data (firmware) to microprocessor 94 so that no conventional operating system is necessary.” (‘676 Patent, col. 16:11-16 (emphasis added).)

“The intrinsic evidence is clear that the form driven operating system is a combination of the TAS firmware **and** forms that **together** provide the sole code to control the microprocessor.” (Defendants’ Opening Br. at 16 (emphasis in Def. Op. Br.).)

“Like the specifications and prosecution histories of the Asserted Patents, Defendants’ construction (i) defines the form driven operating system as a combination of firmware and forms ....” (Defendants’ Opening Br. at 14.)

Given the centrality of the TAS to both sides’ understanding of a form driven operating system (in essence, that TAS plus forms equals form driven operating system), claim construction must provide some meaning to the term TAS. Yet only Cyberfone offers a construction (taken directly from the specification) for TAS that has any substance:

the portion of a form driven operating system that performs at least the two basic functions of (1) generating a template or form from a data stream; and (2) developing a data transaction as the user inputs data in response to prompts in the template or form.

Defendants, on the other hand, offer but a nullity in their proposed construction of TAS, and simply refer back to a form driven operating system (“the firmware component of a form driven operating system”). In other words, having agreed that a form driven operating system is equal

---

<sup>2</sup> See pp. 5-6 below.

to the TAS plus forms, Defendants “define” TAS as the form driven operating system minus forms: “Thus, *rearranged*, the TAS is the ‘form driven operating system without the data streams stored in a form memory.’” (Def. Reply Br. at 12 (emphasis added).) This is not claim construction. This is merely the rearrangement that takes place with basic algebra: if  $A = B + \text{forms}$ , then  $B = A - \text{forms}$ . This shuffling of terms by Defendants to restate the same premise does not teach what either a form driven operating system (the A in the example) or a TAS (the B) does. The intrinsic record explains what these two terms mean, and what roles the form driven operating system and the TAS play in Dr. Martino’s inventions. Claim construction should give meaning to the function and purpose of these terms.

**B. “Form Driven Operating System.”**

Cyberfone’s Proposed Construction	Defendants’ Proposed Construction
Computer code for developing data transactions, which, together with forms, controls the behavior of the microprocessor by logically defining a table of menu options and/or database interfaces. A form driven operating system is not a conventional operating system (such as DOS or Windows).	firmware together with forms that serve as the sole code for controlling a microprocessor, instead of an operating system that runs application programs

**1. The Intrinsic Evidence Compels Cyberfone’s Proposed Construction.**

Cyberfone’s proposed construction (unlike Defendants’) actually defines “form driven operating system,” and does so consistently with the specification. First, the specification makes clear that a form driven operating system is code for developing data transactions:<sup>3</sup>

- “The system of the invention provides for the automatic capture and computerization of data associated with *data transactions* as they occur.” (‘676 Patent, col. 5:50-52 (emphasis added).)
- “entering *data transactions* into databases in accordance with the invention.” (*Id.*, col. 7:9-11, 23-24 (emphasis added).)

<sup>3</sup> See also Def. Op. Br. at 3 (“All four Asserted Patents claim priority to the same 1995 application and disclose devices related to the same technical field: *creating, transmitting, and storing data transactions*.”) (footnote omitted and emphasis added).

- “the creation and storage of a ***data transaction*** in accordance with the invention.” (*Id.*, col. 9:15-17 (emphasis added).)
- “[S]ince ***the data transactions*** are created without the use of an operating system or application programs, the transaction entry device is quite simple and inexpensive and may be readily integrated with the customer’s desktop telephone or portable telephone.” (*Id.*, col. 2:37-42 (emphasis added).)

The centrality of developing data transactions is reflected in the titles of the patents themselves: “Telephone/transaction entry device and system for entering ***transaction data*** into ***databases***” (‘676 and ‘103 Patents) and “***Data transaction*** assembly server” (‘382 Patent).<sup>4</sup> Yet despite the centrality of data transactions in the claimed inventions, Defendants’ constructions do not even mention them. Cyberfone’s proposed construction of form driven operating system, as an initial matter, gives life to the function of the invention, which is intimately related to the generation and communication of data transactions.

Next, while both sides agree that a form driven operating system controls a microprocessor, Cyberfone’s construction explains ***how*** this occurs: “by logically defining a table of menu options and/or database interfaces.” This language also comes directly from the patent specification:

“The TAS firmware and the selected template together control the behavior of the microprocessor 94 ***by logically defining a table of menu options and/or database interfaces*** which are navigated through by the user.” ‘676 Patent, col. 14:5-8 (emphasis added).

This definition, again, gives meaning to the term and helps explain how a form driven operating system works and what makes an operating system “form driven.” Defendants choose to simply ignore what the intrinsic record states regarding how the form driven operating system functions.

---

<sup>4</sup> The ‘024 Patent, which does not have the same two disputed claim terms, is entitled “System for transmission of voice and data over the same communication line.”

Finally, as previously explained, the last part of Cyberfone’s definition – that “a form driven operating system is not a conventional operating system (such as DOS or Windows)” – comes directly from the prosecution history. In his March 18, 1997 Response, the applicant explained that he amended the claims to recite a “form driven operating system” in order to overcome the examiner’s belief that the prior version of the claims “would read on a general purpose computer having a microprocessor running a conventional operating system (such as DOS or WINDOWS) and an application program of the type disclosed by Moriyama . . . .” (*See* Murphy Decl., Ex. E at 18 (‘676 Pat. File History, March 18, 1997 Resp.).) Although Defendants attempt to avoid the intrinsic record on this point, the intrinsic record is clear: a form driven operating system replaces a **conventional** operating system, not any operating system (as Defendants propose in their construction).

Cyberfone’s construction comes directly from the intrinsic record – the specification and the file history – and gives meaning to the term. Only Cyberfone’s construction explains what makes a “form driven operating system” form driven, as opposed to conventional.

In their Reply, Defendants take their complaint regarding Cyberfone’s use of the phrase “together with forms” a step further, suggesting that Cyberfone has put forward a “proposal of ‘data transactions . . . together with forms[.]’” (*See* Def. Reply Br. at 5.) Defendants mischaracterize Cyberfone’s proposed construction. Cyberfone’s actual proposed construction begins: “Computer code for developing data transactions, which, together with forms, controls the behavior of the microprocessor . . . .” It is clear that the computer code is acting together with forms. Cyberfone even made the point repeatedly in its brief: “Although Cyberfone disagrees with Defendants regarding the proper construction of TAS, Cyberfone agrees that TAS plus forms equals the form driven operating system.” (Cyberfone’s Op. Br. at 7 (citing ‘676



patent at col. 14:5-8, 13:50-53, and 14:13-16).) Cyberfone also stated that “the computer code for generating data transactions, in combination with forms, *together* control the behavior of a microprocessor by logically defining a table of menu options and/or database interfaces.”

(Cyberfone’s Op. Br. at 7.) Cyberfone does not see how either Cyberfone’s proposed construction or Cyberfone’s Opening Brief could possibly lead to the conclusion that Cyberfone had proposed “data transactions . . . together with forms.”

Finally, Defendants also contest Cyberfone’s use of the term “computer code” in its proposed construction of form driven operating system. According to Defendants, “the specification consistently associates the TAS portion of the form driven operating system with descriptions such as “firmware,” “microcode,” and “PROM”—limitations not captured by CyberFone’s construction [of TAS].” (Def. Op. Br. at 15.) Given these varied descriptions of the TAS *within the specification itself*, it would be improper to read into the claims a limitation that TAS comprises solely firmware. The patentee never distinguished his inventions from the prior art by describing TAS as firmware (versus, for instance, software). Even in their Reply, Defendants can point to no clear disavowal by the patentee regarding the scope of the term TAS, and thus computer code is the appropriate phrase for this part of the construction of the term form driven operating system. Moreover, Defendants explicitly argued in their opening brief that the TAS is also “computer program code for generating a data transaction.” “Computer program code” is nearly identical to “computer code,” and stands in clear distinction to Defendants’ attempt to read into the term a limitation not present in the claims.

## **2. Defendants’ Construction Is Incomplete, Ignores The Specification, And Misconstrues The Intrinsic Record.**

In their Reply brief, Defendants again attempt to define form driven operating system not by what it is or what it does, but instead by a specific limitation that Defendants hope will allow

them to argue noninfringement. Not only does this narrow and litigation-driven attempt fail to give meaning to the term, but Defendants also fail to describe accurately the characteristic on which they focus.

In short, Defendants argue in their Reply that the inventor disclaimed any operating system running an application. (*See, e.g.*, Def. Reply Br. at 6 (“Both in the patent specification and during the Patent Office prosecution, the patentee repeatedly defined his invention as excluding an operating system that ran applications.”).) This is simply not what the intrinsic record states. For instance, Defendants spend much time selectively quoting (and misquoting)<sup>5</sup> the March 18, 1997 Response in the file history of the ‘676 Patent. Because Defendants repeatedly misconstrue this response, claiming that Dr. Martino distinguished his inventions in ways that he did not, Cyberfone will provide several passages in full. The first passage occurs on pages 18-19 of the response:

“In particular, the Examiners expressed their collective belief that each of the independent claims would read on a general purpose computer having a microprocessor running a conventional operating system (such as DOS or WINDOWS™) and an application program of the type disclosed by Moriyama for presenting forms to the user for completion. In view of this position taken by the Examiners during the interview, Applicant’s undersigned representative agreed to reconsider the claims in view of the cited prior art and to consider amending the claim language to specify that the invention uses a simple form driven operating system in place of the **conventional operating system and application programs**, thereby eliminating much of the overhead and hardware requirements conventionally required in prior art data transaction transactions. The present Supplemental Amendment Response is being submitted to so amend the claims.” (Murphy Decl., Ex. E at 18-19 (‘676 Pat. File History, March 18, 1997 Resp.) (emphasis added).)

Although Defendants ignore the key limitation – that the form driven operating system is being used in place of the *conventional* operating system – the intrinsic record is clear. Later in the

---

<sup>5</sup> Defendants’ quote of the March 18, 1997 response on page 9 of their Reply Brief is simply inaccurate. It omits a lengthy phrase in the middle of the quote, without mention and without using ellipses to signal the omission.

same response, Dr. Martino's attorney also addressed that specific prior art reference

(Moriyama) in greater depth. He explained why claim 1 was not anticipated by Moriyama:

"The system used by Moriyama is a conventional personal computer system which has a hard disc 2 connected to CPU 1 for storing a general purpose management program and the master files and data files necessary for management (see column 3, lines 25-51). Hence, unlike the invention, a personal computer with a **conventional operating system** is used to process a general purpose management program for the entry of data into a database on an associated hard disk. Moriyama nowhere discloses a low cost terminal device which uses a **form driven operating system** instead of **conventional application programs** running on a **standard operating system** to facilitate the entry of data into one or more remote databases (i.e., databases not on an associated disk drive, CD ROM, or other local memory unit). Accordingly, the system disclosed by Moriyama is not particularly relevant to a remote data entry system of the type now claimed." (Murphy Decl., Ex. E at 27 ('676 Pat. File History, March 18, 1997 Resp.) (emphasis added).)

Defendants wish to read "**conventional** operating system" or "**standard** operating system" as "operating system." This is not what the intrinsic record, including the file history, supports. The evidence in the specifications, for example, is overwhelming:

"The microcode of the TAS PROM 95 and the parameter streams from the form/menu memory 96 thus operate together as a simple form driven operating system for microprocessor 94 for all applications and is the sole code used to control microprocessor 94 (i.e., no **conventional operating system** or application programs are provided)." ('676 Patent, col. 13:50-56 (emphasis added).)

"As noted above, the transaction assembly (application) server (TAS) is a data stream stored in TAS PROM 95 which together with the forms from form/menu memory 96 create a simple form driven operating system which provides the necessary control data (firmware) to microprocessor 94 so that no **conventional operating system** is necessary." ('676 Patent, col. 16:11-16 (emphasis added).)

"As will be apparent from the following description, data transaction assembler 18 does not utilize a **conventional operating system** to control the operation of microprocessor 94." ('676 Patent, col. 13:38-41 (emphasis added).)

The objective of the invention is "[e]limination of the requirement of a **conventional operating system** and the associated application programs . . . ." ('676 Patent, 1:61-2:1 (emphasis added).)

"TAS 18 does not utilize a **conventional operating system** to control the processing of application software." ('382 Patent, 16:43-45 (emphasis added).)

Defendants also argue that in attempting to overcome the prior art (Moriyama), Dr. Martino disavowed any application programs regardless of the kind of operating system involved. Again, the actual passages in the prosecution history (copied in full above) do not support such a reading. The form driven operating system of the invention replaced “the conventional operating system and application programs” or “conventional application programs running on a standard operating system.” Defendants ignore the clear intrinsic record. Their proposed construction, which is unsupported by the record, should be rejected.

Finally, Cyberfone previously explained what “i.e.” means, and how it is definitional in the context of the ‘676 Patent specification, with respect to the statement that the form driven operating system “is the sole code used to control microprocessor 94 (i.e., no conventional operating system or application programs are provided).” Defendants’ Reply offers a scattershot response on this basic point, but does not explain what this phrase means if it is not definitional. Defendants again simply wish to ignore the intrinsic record, including portions of the very sentences on which they intend to reply, if those portions of the record do not advance their noninfringement argument. Defendants’ construction, ignoring the role of the *conventional* operating system in understanding a form driven operating system, is unfaithful to the intrinsic record and fundamentally improper.

**C. “Client Module” Of ‘024 Patent.**

Defendants previously admitted that the term “client module” has a plain and ordinary meaning to one of ordinary skill in the art. (See Def. Op. Br. at 18 (“In the software context, a ‘module’ is defined as ‘a program unit that is discrete and identifiable’ or ‘a logically separable part of a program.’”)) (quoting The IEEE Standard Dictionary of Electrical and Electronics Terms (5th ed.1993)).) Defendants also admitted that “claim terms ordinarily take on “the meaning that

a person of ordinary skill in the art would attribute to them . . . .” (See Def. Op. Br. at 11 (internal citations omitted).) Defendants previously argued that this plain and ordinary meaning may be supplanted when “(1) the patentee has chosen to be his own lexicographer, or (2) a claim term lacks such clarity that there is ‘no meaning by which the scope of the claim may be ascertained from the language used.’” (*Id.* (citing *Novartis Pharma. Corp. v. Abbott Labs.*, 375 F.3d 1328, 1334 (Fed. Cir. 2004) (citing *Johnson Worldwide v. Zebco*, 175 F.3d 985, 990 (Fed. Cir. 1999))).) Neither situation applies here. The term “client module” is a term known to persons of ordinary skill in the art – Dr. Martino did not coin it. Nor does the term “lack[] such clarity that there is ‘no meaning by which the scope of the claim may be ascertained.’”

Avoiding these principles, Defendants originally turned to the “relatively narrow” additional interpretative principle of construing claims, if possible, to sustain their validity. See generally Patent Case Management Judicial Guide, Second Edition (2012) 5.2.3.4, at 5-87. As Cyberfone previously explained, this interpretative principle is only employed when a claim term is ambiguous. The Federal Circuit has made this clear:

“In this case, unlike in *Klein* and other cases in which the doctrine of construing claims to preserve their validity has been invoked, the claim term at issue is not ambiguous. Thus, it can be construed without the need to consider whether one possible construction would render the claim invalid while the other would not. The doctrine of construing claims to preserve their validity, a doctrine of limited utility in any event, therefore has no applicability here.”

*Phillips v. AWH Corp.*, 415 F.3d at 1328 (emphasis added). Defendants admit that the term “client module” is not ambiguous. This interpretive principle of last resort does not apply, and the term should be given its plain and ordinary meaning.

Defendants continue to claim, in passing, that applying the plain and ordinary meaning of “client module” in the claims of the ‘024 Patent would encompass matter that “patentee explicitly disclaimed,” and provide a citation to the ‘676 Patent as support. (See Def. Op. Br. at

18 (citing ‘676 Patent, col. 2:58-60); Def. Reply Br. at 11 (citing ‘676 Patent, col. 1:58-60 and March 18, 1997 response in ‘676 Patent File History).) Cyberfone again fails to see how the cited passages in any way support Defendants’ argument. Defendants have not shown that applying the plain and ordinary meaning to the term “client module” would somehow expand the scope of the actual claims of the ‘024 patent to encompass matter that was explicitly disclaimed.

Finally, perhaps recognizing that they have not and cannot show why “client module” should be given the construction they offer, Defendants raise a new argument in their reply brief, invoking the written description requirement of 35 U.S.C. § 112 ¶ 1. This new argument fails for multiple reasons. It is procedurally improper, as Defendants did not raise it in their opening brief. Furthermore, the Court had invited limited briefing on this term to the extent Defendants wished to argue that this term should be construed identically to the term form driven operating system. Moreover, the argument is substantively meritless. Defendants do little more than cite the statute, but to the extent Defendants should ever articulate a challenge based on that ground, Cyberfone will demonstrate that the written description requirement of Section 112 was met.

In sum, the Court should reject Defendants’ request to ascribe the construction of “form driven operating system” from the ‘676, ‘103, and ‘382 Patents to the term “client module” in the ‘024 Patent.

**D. “Transaction Assembly Server (TAS).”**

<b>Cyberfone’s Proposed Construction</b>	<b>Defendants’ Proposed Construction</b>
The portion of a form driven operating system that performs at least the two basic functions of (1) generating a template or form from a data stream; and (2) developing a data transaction as the user inputs data in response to prompts in the template or form.	the firmware component of a form driven operating system

### 1. The Intrinsic Evidence Compels Cyberfone's Construction.

As previously noted, both sides agree that the transaction assembly server or TAS, in combination with forms, together create a form driven operating system. Cyberfone's construction of TAS comes directly from the specification:

The TAS "in general performs the two basic functions of (1) generating a template or form from a data stream and (2) developing a data transaction as the user inputs data in response to prompts in the template or form." ('676 Patent, col. 2:55-60.)<sup>6</sup>

Defendants cannot seriously dispute that Cyberfone's construction accurately describes the functions of the TAS. Indeed, in the portion of Defendants' Opening Brief discussing the term "computer code for generating a data transaction" – which Defendants previously contended was equivalent to TAS – Defendants quoted the specification at length, after noting that the TAS is the code used to generate a data transaction:

- The TAS "in general performs the two basic functions of (1) generating a template or form from a data stream and (2) *developing a data transaction* as the user inputs data in response to prompts in the template or form." ('676 Patent, col. 2:55-60.)
- The TAS firmware of the invention stores the options as well as control programs (microcode) for the processor for use with the templates in *creating the data transactions*. ('676 Pat., col. 6:43-45.)
- Thus, the TAS PROM 95 contains control data (firmware) for the microprocessor 94 and resides in each transaction entry device 12 for *generating a template for a data transaction from a data stream stored in form/menu memory*. ('676 Pat., col. 13:65-14:1.)

(Def. Op. Br. at 22 (emphasis added by Defendants).) Each of these passages supports Cyberfone's construction. Each of these passages describes what the TAS does.

---

<sup>6</sup> Defendants claim that Cyberfone's construction is somehow a "misstatement," apparently because it does not include all of the passage from which it quoted. (Def. Reply Br. at 13-14.) Given the fact that Defendants wish to omit from the construction of TAS all limitations regarding its function whatsoever, this complaint is strange. Moreover, Defendants do not explain how the omitted phrase of the passage – "is absolutely self-contained in its relationship to the hardware of the transaction entry device and in general" – offers any support for their proposed construction of TAS. This phrase does not describe firmware or microcode, for example.

Defendants originally objected that Cyberfone’s construction of TAS contains “superfluous” limitations because, they argued, some of the characteristics of the TAS that Cyberfone delineates are also separately spelled out in claim 16 of the ‘676 Patent. Cyberfone explained that Defendants misapplied *Unique Concepts, Inc. v. Brown*, 939 F.2d 1558 (Fed. Cir. 1991), and that Defendants’ position finds no support in the law.

Defendants apparently have jettisoned that argument, and now simply complain that Cyberfone’s proposed construction of TAS overlaps with its construction of form driven operating system. (See Def. Reply Br. at 13 (asserting that Cyberfone’s construction of TAS “largely replicates” its construction of form driven operating system, and citing *Bicon Inc. v. Straumann Co.*, 441 F.3d 945 (Fed. Cir. 2006) for the proposition that “Claims are to be construed with an eye toward giving effect to all terms in the claim and ‘should not [be] treated as meaningless.’”).) This complaint of Defendants is also meritless. Both parties recognize the interplay of the TAS and the form driven operating system. Cyberfone’s proposed construction, based on the intrinsic record itself, gives life to the function of the TAS and a form driven operating system. Providing a substantive definition to these terms does not render them “meaningless.” Rather, it is Defendants’ attempt to avoid addressing the function of the disputed terms (by defining them circularly) that stands in the way of meaning being ascribed to the terms.

**2. Defendants’ Construction Does Nothing To Define TAS, Ignores The Specification, And Is Inconsistent With Defendants’ Prior Position On The ‘024 Patent.**

Defendants’ Reply Brief makes clear that Defendants do not offer a serious construction for the term TAS. Instead, they explicitly defend their construction by noting that they have rearranged the formula that the TAS plus forms equals a form driven operating system: “Thus, *rearranged*, the TAS is the ‘form driven operating system without the data streams stored in a



form memory.’” (Def. Reply Br. at 12 (emphasis added).) Simply rearranging this formula, by stating that the form driven operating system is the TAS plus forms, and that the TAS is the “form driven operating system without the data streams stored in a form memory,” **without explaining what a form driven operating system does, or what a TAS does**, is not claim construction. It is merely rephrasing the expression of terms that Defendants are unwilling to define. The Court’s claim construction should reflect what the patent itself specifies as what the TAS does and its purpose.

Defendants’ own brief supports Cyberfone’s proposed construction, repeatedly citing supporting passages in the ‘676 specification:

- The TAS “in general performs the two basic functions of (1) generating a template or form from a data stream and (2) **developing a data transaction** as the user inputs data in response to prompts in the template or form.” (‘676 Patent, col. 2:55-60.)
- The TAS firmware of the invention stores the options as well as control programs (microcode) for the processor for use with the templates in **creating the data transactions**. (‘676 Pat., col. 6:43-45.)
- Thus, the TAS PROM 95 contains control data (firmware) for the microprocessor 94 and resides in each transaction entry device 12 for **generating a template for a data transaction from a data stream stored in form/menu memory**. (‘676 Pat., col. 13:65-14:1.)

(Def. Op. Br. at 22 (emphasis added by Defendants).) Each of these passages supports **Cyberfone’s** construction of TAS, which gives actual meaning and definition to the term.

Moreover, in previously arguing that the term “computer program code for generating a data transaction” should be given the same construction as TAS, Defendants explicitly argued that this computer code “is the code that is used to generate a data transaction,” which is the same function as the TAS. (See Def. Op. Br. at 22; *see also id.* (“The identical function of these elements argues in favor of construing the terms to be synonymous.”).) This formulation does **not** limit TAS to firmware. Defendants’ attempt to read into the definition of TAS a limitation

not found in the claims, the specification, or the file history – that TAS must only be firmware – should be rejected.

### III. CONCLUSION

For all of the reasons above, the Court should adopt Cyberfone’s proposed constructions of form driven operating system and transaction assembly server. Finally, the Court should reject Defendants’ request that the term “client module” be found synonymous with the term “form driven operating system.”

Dated: April 18, 2013

Of Counsel:

Marc A. Fenster  
Eric J. Carsten  
Russ, August & Kabat  
12424 Wilshire Boulevard, 12th Floor  
Los Angeles, CA 90025-1031  
(310) 826-7474  
mfenster@raklaw.com  
ecarsten@raklaw.com

BAYARD, P.A.

/s/ Vanessa R. Tiradentes  
Richard D. Kirk (rk0922)  
Stephen B. Brauerman (sb4952)  
Vanessa R. Tiradentes (vt5398)  
222 Delaware Avenue, Suite 900  
Wilmington, DE 19801  
(302) 655-5000  
rkirk@bayardlaw.com  
sbrauerman@bayardlaw.com  
vtiradentes@bayardlaw.com

Attorneys for Plaintiff,  
Cyberfone Systems, LLC